```python
A_CAP = 4
B_CAP = 3
start = (0, 0)
visited = set()
def get_neighbors(state):
    A, B = state
    neighbors = []
    neighbors.append((A_CAP, B))
    neighbors.append((A, B_CAP))
    neighbors.append((0, B))
    neighbors.append((A, 0))
    pour = min(A, B_CAP - B)
    neighbors.append((A - pour, B + pour))
    pour = min(B, A_CAP - A)
    neighbors.append((A + pour, B - pour))
    return neighbors
def bfs():
```

```
Steps to get exactly 2 gallons in 4-gallon jug:
(0, 0)
(4, 0)
(1, 3)
(1, 0)
(0, 1)
(4, 1)
(2, 3)

=== Code Execution Successful ===
```

```
    from collections import deque
    q = deque()
    q.append((start, [start]))
    visited.add(start)
    while q:
        state, path = q.popleft()
        A, B = state
        if A == 2:
            return path
        for nxt in get_neighbors(state):
            if nxt not in visited:
                visited.add(nxt)
                q.append((nxt, path + [nxt]))
    return None
solution = bfs()
print("Steps to get exactly 2 gallons in 4-gallon jug:")
for step in solution:
```